Optimizing Multi-hop Queries in ZigBee Based Multi-sink Sensor Networks

Bing Han[1,2,3] and Gwendal Simon[1]

[1] Institut TELECOM - TELECOM Bretagne, Brest, France
[2] Institut TELECOM - TELECOM ParisTech, Paris, France
[3] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
{bing.han,gwendal.simon}@telecom-bretagne.eu

Abstract. Wireless sensor networks with multiple users collecting data directly from the sensors have many potential applications. An important problem is to allocate for each user a query range to achieve certain global optimality while avoid congesting the sensors in the meanwhile. We study this problem for a ZigBee cluster tree by formulating it into a multi-dimensional multi-choice knapsack problem. Maximum overall query range and max-min fair query range objectives are investigated. Distributed algorithms are proposed which exploit the ZigBee cluster tree structure to keep the computation local. Extensive simulations show that the proposed methods achieve good approximation to the optimal solution with little overhead and improve the network performance.

1 Introduction

Wireless Sensor Networks (WSNs) with multiple mobile sinks can be very useful in emergency applications. A typical example is a monitoring system with some device-equipped firemen gathering data from the fire site in order to determine a safe perimeter, while others operating on the hearth are under real-time alert about risks of nearby explosions. The firemen send requests to and collect data from the sensors within a specific area in multi-hop fashion. Since the firemen are generally interested in what happens nearby, it is beneficial to interact *directly* with the sensors around them instead of via an infrastructure. Under this case, firemen are reasonably considered as *sinks*. Fig. 1 illustrates WSNs with and without an infrastructure. Users in Fig. 1(b) are considered as sinks.

Due to the queries imposed by the sinks, a sensor spends a certain amount of bandwidth for either sending its own data or forwarding data from other sensors. Obviously, if a sensor is impacted by many sinks, it may experience congestion. Packets dropped due to congestion not only waste energy but also degrade query coverage of all related sinks. Thus, it is a natural requirement that each sink sets a proper impact range to avoid congestion and to achieve a global optimality at the same time. Congestion control in WSN has been studied in recent years and proposed solutions have focused on the transport layer [1]. Furthermore, all existing studies have emphasized on providing fairness for sources (sensors) *i.e.*

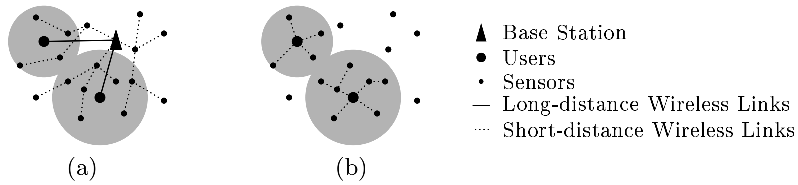

Fig. 1. Multi-sink WSN structures. (a) With infrastructure, data retrieved from base station. (b) Infrastructure-less, data retrieved directly from sensors.

allocating for each sensor a fair amount of bandwidth. In contrast, we investigate both maximality and fairness objectives *in favor of sinks*, *i.e.* allocating for each sink a proper query impact range to achieve global optimality while not congesting the sensors. We emphasize that doing so has practical significance. When sinks are independent users of the WSN, we usually would like to give each of them a fair chance to access the network thus prevent certain users from being starved. When the users act towards the same goal, one may want to maximize the sum of their impact range. We also argue that whole-network query coverage on the sensors, as a usually studied problem, is less important under the aforementioned application scenario. Instead, data from sensors within a reasonable query range should be reported with high reliability. Thus, sensors far away from all sinks may have no query on them, as shown in Fig. 1(b).

We investigate this *impact range allocation problem* and confine our study with the following assumptions: (i) Each fireman tends to set its impact range as large as possible in order to maximize individual security. The impact range is measured by hop numbers such that a k hop impact range will cover all k hop neighbors of the sink. (ii) ZigBee network will be employed since it is especially suitable for low power, low rate wireless sensor networks and supported by many off-the-shelf WSN products. Because energy supply is usually scarce in WSN, cluster tree mode of the ZigBee will be considered. (iii) No in-network data aggregation or compression. (iv) We will investigate two global optimization objectives: maximizing the overall impact range and allocating max-min fair impact range for the sinks although other objectives may be applied.

The contributions of this study are three-fold. Firstly, the impact range allocation problem is formulated and studied as a Multi-dimensional Multiple choice Knapsack Problem (MMKP) with two optimization objectives. Secondly, distributed heuristic is proposed based on solving a local optimization problem on congested sensors. Finally, simulation results show that the proposed algorithm obtains a good approximation to the optimal solution and is able to alleviate congestion therefore improve network performance. The rest of the paper is organized as follows: Related works are briefly surveyed in Section 2. ZigBee standard is briefly introduced and a multi-sink WSN architecture is proposed in Section 3. In Section 4, we formally describe the proposed problems. Distributed algorithms are discussed in Section 5 and evaluated by simulations in Section 6.

Some perspectives of these first results may deserve further investigation. We discuss them and conclude the paper in Section 7.

2 Related Works

We will formulate the impact range allocation problem as an MMKP in Section 4. For detailed descriptions of MMKP, please refer to [2]. In general, MMKP can be solved with branch and bound strategy and there are several off-the-shelf Mixed Integer Programming (MIP) solvers available, *e.g.* the GLPK package[3]. Besides, various algorithms have been proposed to solve the MMKP either exactly or approximately. Due to the NP-completeness of the problem, only approximate algorithms are suitable for large or/and online problems [4,5,6]. In particular, an MMKP with a single constraint degrades to a Multiple Choice Knapsack Problem (MCKP). The lexicographic Max-Min Fairness (MMF) is a generalization of the traditional max-min fairness defined in [7]. They are equivalent on convex solution sets but MMF solution exists on general sets as well. We will employ the former because the problem has discrete parameter values. MMF has been used in formulating various resource allocation problems in the networking area and general MMF concepts and formal problem formulations, algorithms as well as example design problems can be found in [8].

Similar query allocation problem has been studied based on queries with continuous radius in our previous work [9]. We investigate hop-based query in the current paper. Under this discrete setting, the problem becomes combinatoric which is harder and deserves methods other than those employed in [9].

Concluding the discussions above, our MMKP formulation of the maximum overall impact range problem is exactly the same MMKP problem mentioned above and can be solved by those algorithms. So we emphasize on formulating the MMKP problem with MMF as an objective which is novel. Furthermore, we will consider solving both problems in a distributed way in contrast to the centralized algorithms already proposed.

Note also that our work is based on the ZigBee cluster tree operating mode which necessitates a beacon enabled 802.15.4 network. Beacon scheduling, beacon period and superframe length have great impact on the performance of the network, as investigated in [10,11,12]. In our study, we will focus on solving a distributed optimization problem, thus we schedule the beacons only between parents and children and fix beacon period and superframe length.

3 ZigBee Based Wireless Sensor Networks

The ZigBee specification [13] defines addressing, network maintenance and routing for unicast, multicast or broadcast packets at network layer for LR-WPANs. It specifies IEEE 802.15.4 [14] as its MAC layer, which provides sleep mode feature based on superframes bounded by the beacons. This feature is available only in the synchronized tree network. A ZigBee network is initiated and controlled by a single ZigBee coordinator, while other devices act as either a router, if they participate in routing, or end devices otherwise. The routing mechanism employed by ZigBee combines the flat routing known as AODV [15] and the hierarchical routing based on the cluster tree. When the network operates in ad hoc mode, the data is generally delivered by AODV and only when AODV routing fails, the cluster tree routing is used. The network can also operate in pure cluster tree mode where only tree routing is used. Under this case, a router only need to decide to forward a packet to either its parent or one of its children based on its own address, destination address of the packet, the maximum children number (Cm), maximum router number (Rm) and maximum network depth (Lm). The last three parameters are pre-configured for a given ZigBee network and are known to every device. Cluster tree routing does not employ any route discovery message thus is more appealing for energy constrained networks such as WSNs. Besides, a sink (user) in a ZigBee-based WSN may act as either end devices or routers. If mobility is considered, the movement of a sink router may result in reconstruction of the subtree rooted at itself. Thus we consider only the case that sinks are end devices in this paper.

In order to collect data from the sensors, a sink sends a query to all sensors within a certain number of hops around it. All queried sensors send data back at the required rate until modified by another query. The queries are sent via hop-bounded flooding on the tree and data is sent back via unicast cluster tree routing, both supported by the ZigBee specification. Especially for the latter, we assume sensors send a copy of the same data to each querying sink rather than using multicast. Fig. 2 illustrates an example multi-sink WSN with two sinks and a query at radius of 3 hops.

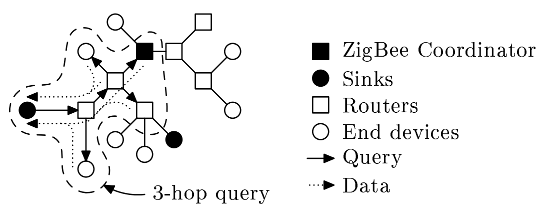

Fig. 2. Multi-sink WSN based on ZigBee tree structure.

4 Model and Problem Formulation

We consider a set $\mathcal{V}$ of n sensors and a set $\mathcal{M}$ of m sinks. The communication tree employed by ZigBee is defined as $T = (\mathcal{V} \cup \mathcal{M}, E)$ with $\mathcal{V} \cup \mathcal{M}$ the node set and E the link set. There is a link between two nodes of $\mathcal{V} \cup \mathcal{M}$ if they have a parent-child relationship between them. The available bandwidth r_i of a sensor is assumed to be the shared bandwidth seen by the application and we assume all r_is take the same value. A **query** of sink p is disseminated to all sensors within

$u(p)$ hops on the tree from p. $u(p)$ is also referred to as the impact range of query p. Sensors under query will generate data at a certain constant rate in response. Because the data are routed back to the sink along the communication tree, the amount of bandwidth a sensor i has to spend, as a result of the impact of p, is equal to or larger than its upstream sensors along the route. This holds as long as we assume there is no compression or aggregation on the data. A **configuration** is a set of impact range chosen by all sinks, noted as $C = \{u(p) : \forall p \in \mathcal{M}\}$. We say that a configuration is *feasible* when the bandwidth required to handle the queries on each sensor is less than its available bandwidth. The set of feasible configurations will be referred to as $\mathcal{C}$.

The impact range allocation problem is to find a subset of $\mathcal{C}$ which achieves certain optimization objectives. We formulate it into a generalized MMKP. Let each possible impact range of sink p correspond to an item to be selected and the value of the items is the hop-distances in T, then we have $u(p) \in [1, d_T]$ where d_T is the diameter of T. Since each sink sets its impact range to a particular value at a certain time, the items can be seen as grouped into m classes each corresponding to a certain sink. A binary variable x_{pu} is then associated with sink p where $x_{pu} = 1$ indicates that $u(p) = u$ and $x_{pu} = 0$ otherwise, with $u \in [1, d_T]$. The bandwidth provided by a sensor i to a sink p when p takes impact range level u is denoted as r_{ipu} and is mapped to the ith dimension of weight of item u in class p. Each sensor forms a constraint dimension with its available bandwidth r_i. The general MMKP is formulated as follows:

$$\begin{aligned} &\text{Achieve:} && \text{General Objective} && \\ &\text{Subject to:} && \textstyle\sum_{p \in \mathcal{M}} \sum_{u=1}^{d_T} r_{ipu} x_{pu} \le r_i,\ i \in \mathcal{V} && (1) \\ & && \textstyle\sum_{u=1}^{d_T} x_{pu} = 1,\ p \in \mathcal{M} && (2) \\ & && x_{pu} \in \{0, 1\},\ p \in \mathcal{M},\ u \in [1, d_T] && (3) \end{aligned}$$

We propose two explicit objectives in place of the general one: (i) to maximize the sum of impact range of all sinks *i.e.* $\sum_{p \in \mathcal{M}} u(p)$, noted as **Maximum Impact Range** (**MMKP-MIR**) problem, and (ii) to find **Max-Min Fair** impact range allocations (**MMKP-MMF**). Searching for MMF solutions in discrete solution space has been proven to be NP-hard [16]. Similarly, we can also prove MMKP-MIR is NP-hard but this is considered as technical thus omitted.

5 Distributed Algorithms

The basic idea of the distributed algorithm is to solve a smaller local problem at the congested sensor based on its estimation of the potential traffic. Algorithms for MMKP-MMF and MMKP-MIR are similar thus are put under a uniform algorithmic framework.

5.1 Uniform Algorithmic Framework

At the sink side, the algorithm starts with a *slow start* phase. As shown in Algorithm 1, function `initLevel` increase the query level with exponential growth to quickly discover a potentially congested sensor. On the sensors, the `congested` function detects a potential congestion state. We propose using the collision intensity information obtained from MAC layer as a simple congestion identification mechanism. Accurate congestion detection methods can be applied but they are out of scope of this paper. On congestion, function `solveMCKP` calculates an impact range allocation for the related sinks. We exploit GLPK for MMKP-MIR or a heuristic we will propose for MMKP-MMF in the simulation implementation. Note that GLPK is obviously infeasible for an implementation on real sensor devices, we use it the simulations only for simplicity. Then the congested sensor sends an `adjust-level` message to the related sinks with the local solution. On receiving an `adjust-level` message, the sink sets its impact range to the suggested level if the suggested level is smaller than the current value. Then the `increaseLevel` function increases the impact level linearly because the suggested impact range is already near the optimal.

In the following, we first present the traffic estimation mechanism, then we propose a heuristic for the MMF version of the local problem, finally, we explain why linear increment in `increaseLevel` of the impact level is necessary.

Algorithm 1: Distributed Heuristic

```
Sink Part   : Run at sink p
send < level, p, 1 >
while no < adjust-level > message do
    level ← initLevel()
    send < level, p, level >
level ← adjustLevel()
while true do
    while no < adjust-level > message do
        level ← increaseLevel()
        send < modify-level, p, level >
    level ← adjustLevel()
    send < modify-level, p, level >

Sensor Part: Run at sensor i
while true do
    if congested() then
        C ← solveMCKP()
        for ∀p : l_p ∈ C do
            send < adjust-level, l_p > to p;
```

Algorithm 2: Local MCKP-MMF

```
input  : S ⊆ M, U, h, h_p, d, r_i
output: MCKP-MMF configuration C
for p ∈ S do
    C ← {S_p = (level_p ← 0, state_p ← active)}
for u_m ∈ U, m ← 1 to |U| do
    A ← {S_p : state_p = active}
    if A = ∅ then break
    equclass ← (d − h + h_p)/2
    sort(A, Δ_T(equclass, h_p, u_m))
    for a_j ∈ A, j ← 1 to |A| do
        C' ← C, level'_{a_j} ← u_{m+1}
        if feasible (C') then
        level_{a_j} ← u_{m+1}
        else
            for a_k ∈ A, k ← j to |A| do
                state_{a_k} ← stop
            break
return C
```

5.2 Traffic Estimation

The aim of traffic estimation is to provide information about the additional traffic load offered on a certain router if the impact range of a query becomes one hop larger. We propose a local estimation profiting the ZigBee cluster tree structure, instead of measuring the real traffic.

Consider the ZigBee tree T rooted at the ZigBee coordinator, as shown in Fig. 3(a). For the querying sink s, the whole network can be seen as a tree T_s rooted at itself as in Fig. 3(b). Thus for a router (the ZigBee Coordinator is considered also as a router), the additional traffic passing through it comes from the to-be-covered devices that are his descendent on T_s. We assign a label (h, h_s) to each device in the network, where h and h_s denote the depth of the device on

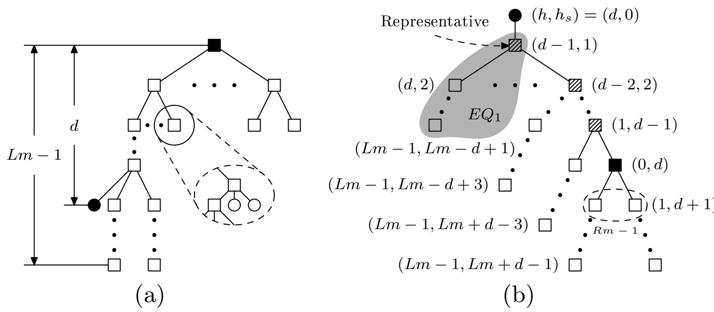

Fig. 3. Different views of a ZigBee routing tree. (a) Tree T, rooted at the ZigBee coordinator. (b) Tree T_s, rooted at the sink, labels assigned to devices.

T and T_s, respectively. Note that every device knows h on joining the network and h_s is actually its hop distance to s which could be obtained from the query messages. Then we classify the routers whose label satisfies $h - h_s = d - 2i$ into equivalent classes EQ_i, and the router with $h_s = i$ is referred to as the *representative router* of the equivalent class. As shown in Fig. 3(b), the routers belong to EQ_1 are grouped into a shadowed area.

Now consider a router in $r \in EQ_i$ with label (h, h_s), let $\Delta_R(i, h_s, k)$ be the number of additional routers and $\Delta_E(i, h_s, k)$ the number of additional end devices that will be handled by r when the impact range of the query increases from k to $k + 1$. If every device sends data at constant rate B, the additional traffic load $\Delta_T(i, h_s, k)$ on router r should be:

$$\Delta_T(i, h_s, k) = B\left(\Delta_R(i, h_s, k) + \Delta_E(i, h_s, k)\right). \tag{4}$$

We first derive $\Delta_R(i, h_s, k)$, then $\Delta_E(i, h_s, k)$ can be obtained as:

$$\Delta_E(i, h_s, k) = \Delta_R(i, h_s - 1, k - 1)(Cm - Rm). \tag{5}$$

Let Δ_R^n and Δ_R^r be the Δ_R functions of non-representative and representative routers respectively. Then we have:

$$\Delta_R^n(i, h_s, k) = \begin{cases} Rm^{k-h_s+1}, & \text{if } h_s - 1 \leq k \leq Lm - d + 2i - 2 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $1 \leq i \leq d$ and $i + 1 \leq h_s \leq Lm - d + 2i - 1$, and for a representative router, which implies $h_s = i$, we have:

$$\Delta_R^r(1, 1, k) = \begin{cases} Rm\Delta_R^n(1, 2, k) + \Delta_R^r(2, 2, k), & \text{if } 0 < k \leq Lm + d - 2 \\ 1, & \text{if } k = 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$\Delta_R^r(i, i, k) = \begin{cases} (Rm - 1)\Delta_R^n(i, i + 1, k) & \\ \quad + \Delta_R^r(i + 1, i + 1, k), & \text{if } i - 1 < k \leq Lm - d + 2i - 2 \\ 1, & \text{if } k = i - 1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $2 \leq i \leq d$.

Remark: the estimation above is an upper bound of the traffic and it is accurate only when the network address is fully used by the devices, which is hardly true in practice. Therefore, we propose to apply a scaling parameter on the estimation to take address utilization into account, *e.g.* the number of children devices.

5.3 Local MCKP Solution

Various existing algorithms can be applied to solve an MCKP with maximum impact range objective (MCKP-MIR). We will solve it with GLPK and for a more practical implementation, a light weight heuristic algorithm could be applied. We propose a heuristic for the MMF local problem (MCKP-MMF) here.

MCKP-MMF heuristic starts at a trivial configuration $C = (0, \ldots, 0)$ with all sinks marked as 'active', then discovers a partial feasible solution by greedily increasing the impact range of the sink that generates the least extra traffic, round by round. At each round, the active sinks are sorted by their extra traffic load estimated by Δ_T at the corresponding impact level in ascending order. Then the algorithm increases their impact levels by one, one sink after another, from the least costly sink to the most costly one. Similar idea based on the *savings* has been used in [4]. If the constraint is violated at a certain round, the first sink that violates the constraint and all sinks after it are marked as 'stopped'. The algorithm terminates once all sinks are marked as 'stopped'.

Algorithm 2 describes this heuristic. The input $\mathcal{S}$ of the algorithm is a subset of the sinks which have query on sensor i, as each sensor records each query it is handling, $\mathcal{S}$ is known to i. Parameter h and h_p are the depth of sensor i on tree T and tree T_p, respectively. The `sort` function in the algorithm sorts the active sinks in A in ascending order with Δ_T as keys.

5.4 Dynamic Impact Range Adaption

A sink may receive multiple notifications from multiple congested sensors. In order to satisfy the most stringent constraint, it needs to adjust its impact range only when the new range is lower than its current one. The side effect of this policy is the impact range tends to decrease in the long run and a sink may not be able to know its optimal impact range. To help the sinks to jump out of a potential local optimal assigned by Algorithm 2, each sink tries to increase its impact level periodically. Thus, `increaseLevel` is employed after the sink adjusts its impact range. Similar effects have been observed and the same countermeasure has been employed in [9].

6 Evaluation

We implemented the algorithms and the basic functionalities of ZigBee network layer on top of IEEE 802.15.4 implementation [17] in ns2 [18]. The algorithm is evaluated with the metrics defined in Table 1. The simulation parameters are summarized in Table 2. The size of the network is chosen to include a small

Table 1. Evaluation metrics.

γ	=	$\frac{\text{Query data receiving rate at sinks (bps)}}{\text{Query data sending rate (bps)}}$
T_{app}	=	Query data receiving rate at sinks (bps)
O_{mac}	=	$\frac{\text{MAC control message sending rate (bps)}}{\text{Query data receiving rate (bps)}}$
O_{app}	=	$\frac{\text{Application control message sending rate (bps)}}{\text{Query data receiving rate (bps)}}$
I	=	$\frac{\text{sum of impact level of sinks}}{\text{number of sinks}}$, at a certain simulation time.
I^*		same as I, obtained by an exact algorithm.
FI	=	$\frac{\left(\sum_{i=1}^{m} x_i\right)^2}{m \sum_{i=1}^{m} x_i^2}$, m: number of queries; x_i: impact range of query i

Table 2. Simulation parameters.

Node distribution	Uniform
Topology	50 nodes at 100m×100m 100 nodes at 140m×140m
Number of sinks	4% number of the nodes, randomly chosen
MAC layer	IEEE 802.15.4, $BO = SO = 6$, $Lm = 10$, $Cm = 3$, $Rm = 3$
NWK layer	ZigBee cluster tree routing
Wireless Tx/Rx	15m, two ray ground model, omniscient antenna
Bandwidth	250kbps at 2.4GHz band
Query data rate	Light load: 100bps, Heavy load: 800bps
Simulation time	Query starts: 70s, query stops 300s, simulation stops: 350s

and a large network. We also select two representative values of the required data rate to simulate a light load network and a heavy load network. During the simulation, ZigBee network formation takes about 60 seconds so we start the queries at 70 seconds.

6.1 Data Arrival Ratio and Throughput

The effectiveness of the two algorithms in controlling the congestion is shown by comparing the query data arrival ratio against a network without impact range allocation algorithms, where the sinks take the maximum allowed impact range. From Fig. 4, we observe that the uncontrolled network delivers the data at a relatively lower ratio than the controlled network. For a heavy load network, less of congestion control is generally detrimental: arrival ratios can be as low as about 0.37, which is unacceptable. In Fig. 5, we present the aggregated throughput of the query data. Under congestion, the throughput could be even lower, as observe in the heavy load cases in Fig. 5.

6.2 Control Message Overhead

We are also interested in how much it costs to achieve query control with the distributed algorithm. Both the MAC layer and the application layer overhead are investigated. At MAC layer, the overhead comes from the beacon, acknowledgment and ARP requirement/response messages, while at application layer, it comes from the query message and the adjust-level message as described in Algorithm 1. Note that there is no routing overhead in the network layer and the network formation overhead is negligible since it is done only at the beginning of the network operation period and we do not consider network re-formation. Fig. 6 shows these results. The values shall be interpreted as how many bits of control message is needed to successfully deliver one bit of query data. At MAC layer, advantages of impact range control are obvious under high load case as the overhead is comparable or even lower than that of uncontrolled network. At

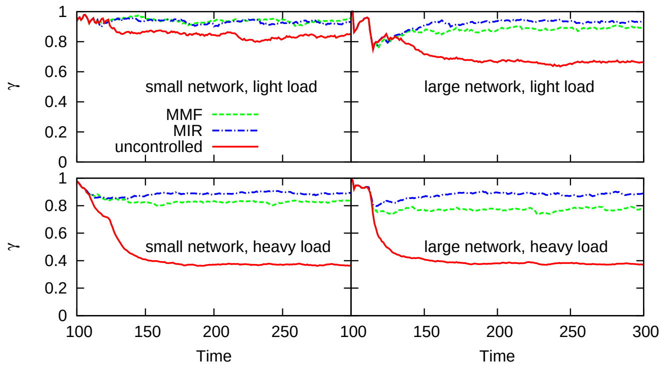

Fig. 4. Data arrival ratio.

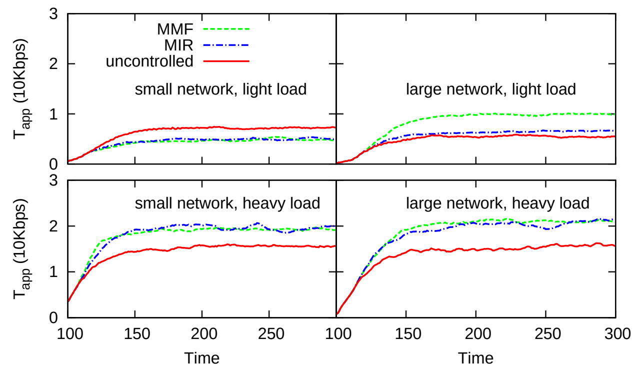

Fig. 5. Aggregated data throughput.

application layer, the uncontrolled networks always have negligible overhead. In contrast, the control overhead goes abruptly for networks with either MIR or MMF when the query load is light. However, the overhead keeps at a very low level under heavy load cases.

6.3 Impact Range and Fairness Index

The average impact range of the 4 sinks in the large network, obtained by both the distributed algorithms and an external problem solver is plotted in Fig. 7. The external problem solver obtains an *optimal* solution for both MIR and MMF with the traffic information traced from the uncontrolled network. We see that for MMF case, the approximation is quite near to the optimal. While for MIR in a light load network, the optimal is further above what the distributed algorithm can achieve. This is because the 'select the minimum' strategy used in Algorithm 1 gives a sink more chances to follow a smaller impact range. The problem becomes less obvious when the local solution gives sinks similar impact

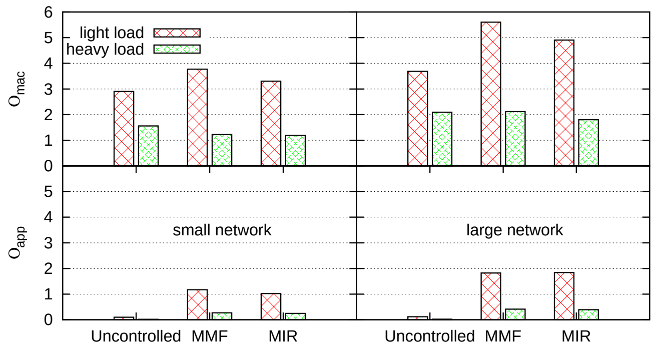

Fig. 6. Control message overhead.

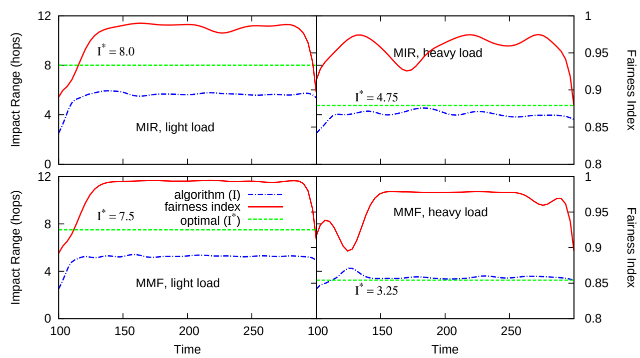

Fig. 7. Impact range and fairness index.

ranges. We observe also that the heuristic over-performs the optimal under heavy loaded MMF case. This is because when the local solution suggested by the congested sensor is already very near the optimal, the linear increment procedure of the sink may generate transient congestion state in the network. We leave these problems for future works. Concerning the fairness index, MMF performs slightly better, as can be expected.

7 Conclusion

We have proposed and solved the impact range allocation problem for a WSN based on ZigBee tree. Optimizing such networks in favor of the sinks (users) has practical significance. MMF and MIR, two commonly used optimization objectives have been studied in the paper and they conform with the MMKP formulation. Distributed algorithms have been proposed to solve the problems through cooperation between sinks and sensors. By exploiting the ZigBee cluster tree structure, the computation was done fully locally. Simulation results have shown that the proposed algorithms perform well in congestion control with little overhead. They especially efficient in large networks with heavy queries.

As future works, it will be interesting to cope with dynamic networks, mobility of sinks is a challenge on the way to a realistic network. On the other hand, although only bandwidth is considered in this study, the problem formulation and the algorithms may be adapted to cope with other types of resources, for example, energy, storage, computation power, *etc.*

References

1. Wang, C., Sohraby, K., Li, B., Daneshmand, M., Hu, Y.: A survey of transport protocols for wireless sensor networks. IEEE Network **20**(3) (2006) 34–40
2. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc., New York, NY, USA (1990)
3. `http://www.gnu.org/software/glpk/`
4. Khan, S., Li, K.F., Manning, E.G., Akbar, M.M.: Solving the knapsack problem for adaptive multimedia systems. Stud. Inform. Univ. **2**(1) (2002) 157–178
5. Hifi, M., Michrafy, M., Sbihi, A.: Heuristic algorithms for the multiple-choice multi-dimensional knapsack problem. Journal of the Operational Research Society **55**(12) (2004) 1323–1332
6. Hifi, M., Michrafy, M., Sbihi, A.: A reactive local search-based algorithm for the multiple-choice multi-dimensional knapsack problem. Computational Optimization and Applications **33**(2-3) (2006) 271–285
7. Radunovic, B., Boudec, J.Y.L.: A Unified Framework for Max-Min and Min-Max Fairness with Applications. ACM/IEEE Trans. on Networking **15**(5) (2006) 1073–1083
8. Ogryczak, W., Pioro, M., Tomaszewski, A.: Telecommunications network design and max-min optimization problem. Journal of Telecom. and Information Tech. **3** (2005) 43–56
9. Han, B., Simon, G.: Fair capacity sharing among multiple sinks in wireless sensor networks. In: Proc. of the IEEE MASS Conf. (Oct. 2007) 1–9
10. Yeh, L.W., Pan, M.S., Tseng, Y.C.: Two-way beacon scheduling in zigbee tree-based wireless sensor networks. In: The IEEE Int. Conf. on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC'08). (2008)
11. Pan, M.S., Tseng, Y.C.: Quick convergecast in zigbee beacon-enabled tree-based wireless sensor networks. Comput. Commun. **31**(5) (2008) 999–1011
12. Kohvakka, M., Kuorilehto, M., Hännikäinen, M., Hämäläinen, T.D.: Performance analysis of ieee 802.15.4 and zigbee for large-scale wireless sensor network applications. In: PE-WASUN'06: Proc. of the 3rd ACM Int. workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks. (2006) 48–57
13. ZigBee Specification, Document 053474r17 (2008)
14. IEEE 802.15.4 Standard (2006)
15. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: Proc. of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99). (1999) 90–100
16. Sarkar, S., Tassiulas, L.: Fair allocation of discrete bandwidth layers in multicast networks. In: Proc. of the IEEE INFOCOM Conf. (Mar. 2000)
17. Zheng, J., Lee, J.M.: Low rate wireless personal area networks - ns2 simulation platform. `http://www-ee.ccny.cuny.edu/zheng/pub/`
18. `http://www.isi.edu/nsnam/ns/`